

# HW6\_Q1

November 1, 2019

```
[15]: from gurobipy import *
```

## 1 Q1

```
[16]: f = open("shortest_path_data-1.txt", "r")
lines = f.readlines()[1:]
myModel = Model("Shortest_path")
cost = [[0 for i in range(8)] for j in range(8)]
myVars = [[0 for i in range(8)] for j in range(8)]
for i in lines:
    initial = i.split()
    origin = int(initial[0])
    destination = int(initial[1])
    cost[origin - 1][destination - 1] = float(initial[2])
print(cost)
```

```
[[0, 1.0, 2.0, 0, 0, 0, 0, 0], [0, 0, 1.0, 5.0, 2.0, 0, 0, 0], [0, 0, 0, 2.0,
1.0, 4.0, 0, 0], [0, 0, 0, 0, 3.0, 6.0, 8.0, 0], [0, 0, 0, 0, 0, 3.0, 7.0, 0],
[0, 0, 0, 0, 0, 0, 5.0, 2.0], [0, 0, 0, 0, 0, 0, 0, 6.0], [0, 0, 0, 0, 0, 0, 0,
0]]
```

```
[17]: for i in range(8):
        for j in range(8):
            curVar = myModel.addVar(vtype = GRB.CONTINUOUS, name = "X" + str(i) +
→str(j))
            myVars[i][j] = curVar
myModel.update()
```

```
[18]: objExpr = LinExpr()
for i in range(8):
    for j in range(8):
        curVar = myVars[i][j]
        objExpr += cost[i][j] * curVar

myModel.setObjective(objExpr, GRB.MINIMIZE)
```

```
print(objExpr)
```

```
<gurobi.LinExpr: 0.0 X00 + X01 + 2.0 X02 + 0.0 X03 + 0.0 X04 + 0.0 X05 + 0.0 X06  
+ 0.0 X07 + 0.0 X10 + 0.0 X11 + X12 + 5.0 X13 + 2.0 X14 + 0.0 X15 + 0.0 X16 +  
0.0 X17 + 0.0 X20 + 0.0 X21 + 0.0 X22 + 2.0 X23 + X24 + 4.0 X25 + 0.0 X26 + 0.0  
X27 + 0.0 X30 + 0.0 X31 + 0.0 X32 + 0.0 X33 + 3.0 X34 + 6.0 X35 + 8.0 X36 + 0.0  
X37 + 0.0 X40 + 0.0 X41 + 0.0 X42 + 0.0 X43 + 0.0 X44 + 3.0 X45 + 7.0 X46 + 0.0  
X47 + 0.0 X50 + 0.0 X51 + 0.0 X52 + 0.0 X53 + 0.0 X54 + 0.0 X55 + 5.0 X56 + 2.0  
X57 + 0.0 X60 + 0.0 X61 + 0.0 X62 + 0.0 X63 + 0.0 X64 + 0.0 X65 + 0.0 X66 + 6.0  
X67 + 0.0 X70 + 0.0 X71 + 0.0 X72 + 0.0 X73 + 0.0 X74 + 0.0 X75 + 0.0 X76 + 0.0  
X77>
```

```
[19]: constExpr = LinExpr()  
for j in range(8):  
    if cost[0][j] != 0:  
        constExpr += 1 * myVars[0][j]  
myModel.addConstr(lhs = constExpr, sense = GRB.EQUAL, rhs = 1)  
  
constExpr = LinExpr()  
for i in range(8):  
    if cost[i][7] != 0:  
        constExpr += 1 * myVars[i][7]  
myModel.addConstr(lhs = constExpr, sense = GRB.EQUAL, rhs = 1)  
  
for i in range(1,7):  
    constExpr = LinExpr()  
    for j in range(8):  
        if cost[i][j] != 0:  
            constExpr += 1 * myVars[i][j]  
        if cost[j][i] != 0:  
            constExpr -= 1 * myVars[j][i]  
    myModel.addConstr (lhs = constExpr, sense = GRB.EQUAL, rhs = 0)  
  
myModel.update
```

```
[19]: <bound method Model.update of <gurobi.Model Continuous instance Shortest_path: 0  
constrs, 64 vars, Parameter changes: LogFile=gurobi.log, CSIdleTimeout=1800>>
```

```
[20]: myModel.write(filename = "Shortest_path.lp")  
myModel.optimize()  
print("Optimal Objective: \n" + str(myModel.ObjVal))  
print("Optimal Solution:")  
allVars = myModel.getVars()  
for curVar in allVars:  
    print(curVar.varName + " " + str(curVar.x))
```

Optimize a model with 8 rows, 64 columns and 32 nonzeros  
Coefficient statistics:

Matrix range [1e+00, 1e+00]  
 Objective range [1e+00, 8e+00]  
 Bounds range [0e+00, 0e+00]  
 RHS range [1e+00, 1e+00]  
 Presolve removed 2 rows and 52 columns  
 Presolve time: 0.01s  
 Presolved: 6 rows, 12 columns, 24 nonzeros

Iteration	Objective	Primal Inf.	Dual Inf.	Time
0	3.9920000e+00	1.503000e+00	0.000000e+00	0s
3	8.0000000e+00	0.000000e+00	0.000000e+00	0s

Solved in 3 iterations and 0.02 seconds

Optimal objective 8.000000000e+00

Optimal Objective:

8.0

Optimal Solution:

X00 0.0  
 X01 1.0  
 X02 0.0  
 X03 0.0  
 X04 0.0  
 X05 0.0  
 X06 0.0  
 X07 0.0  
 X10 0.0  
 X11 0.0  
 X12 1.0  
 X13 0.0  
 X14 0.0  
 X15 0.0  
 X16 0.0  
 X17 0.0  
 X20 0.0  
 X21 0.0  
 X22 0.0  
 X23 0.0  
 X24 1.0  
 X25 0.0  
 X26 0.0  
 X27 0.0  
 X30 0.0  
 X31 0.0  
 X32 0.0  
 X33 0.0  
 X34 0.0  
 X35 0.0  
 X36 0.0

X37 0.0  
X40 0.0  
X41 0.0  
X42 0.0  
X43 0.0  
X44 0.0  
X45 1.0  
X46 0.0  
X47 0.0  
X50 0.0  
X51 0.0  
X52 0.0  
X53 0.0  
X54 0.0  
X55 0.0  
X56 0.0  
X57 1.0  
X60 0.0  
X61 0.0  
X62 0.0  
X63 0.0  
X64 0.0  
X65 0.0  
X66 0.0  
X67 0.0  
X70 0.0  
X71 0.0  
X72 0.0  
X73 0.0  
X74 0.0  
X75 0.0  
X76 0.0  
X77 0.0

[ ]: